

n 维空间随机矩阵变换的音频置乱算法

李环^{1,2}, 覃征³, 张选平¹, 邵利平¹

(1. 西安交通大学电子与信息工程学院, 710049, 西安; 2. 东莞理工学院计算机学院, 523808, 广东东莞;
3. 清华大学计算机科学与技术系, 100083, 北京)

摘要: 针对一维线性序列置乱方法加密参数少、特殊矩阵变换易受攻击等问题,提出一种 *n* 维空间随机矩阵变换的音频置乱算法. 引入 *m* 进制转换的思想,将音频序列采样点值进行变维操作,再通过随机变换阵进行矩阵变换完成置乱. 算法的音频置乱恢复无需计算最小可恢复周期,仅通过随机变换阵在 Z_m 上的逆阵进行一性次恢复即可. 实验结果表明,算法在任意 *n* 维空间均具有很好的置乱效果,置乱和恢复代价较低,鲁棒性较强,能抵抗添加高斯白噪声、滤波、任意剪切、缩放音频值等攻击.

关键词: 音频置乱;随机矩阵变换;*m* 进制;*n* 维空间

中图分类号: TP391 **文献标志码:** A **文章编号:** 0253-987X(2010)04-0013-05

An *n*-Dimensional Space Audio Scrambling Algorithm
Based on Random Matrix

LI Huan^{1,2}, QIN Zheng³, ZHANG Xuanping¹, SHAO Liping¹

(1. School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China; 2. Department of Computer Science, Dongguan University of Technology, Dongguan, Guangdong 523808, China; 3. Department of Computer Science and Technology, Tsinghua University, Beijing 100083, China)

Abstract: It is known that the conventional audio scrambling methods based on one-dimensional linear mapping have less encrypting parameters and are easy to be attacked. A new *n*-dimensional space audio scrambling algorithm based on random matrix is proposed to address these problems. The idea of *m*-ary is introduced and the audio value of each position coordinate is transformed into different dimensional space. The scrambling is completed by the high dimensional random matrix. The proposed algorithm doesn't need to compute the least recovering period, and only uses the inverse of the random matrix in Z_m to recover the scrambled audio in real time. Experimental results show that the proposed algorithm has a wonderful one time scrambling performance in arbitrary *n*-dimensional space with low scrambling cost and recovering costs. The algorithm is robust in enduring common attacks, such as Gaussian noise addition, data loss, data amplification and data reduction.

Keywords: audio scrambling; arbitrary matrix; *m*-ary; *n*-dimensional space

置乱技术可应用于信息隐藏、信息分存和数字水印的预处理或后处理^[1-2],已有的图像置乱方法有 Arnold 变换、幻方变换、Hilbert 曲线、Tangram 算法、IFS 模型、Conway 游戏和 Gray 码等^[3]. 数字音

频为一维序列,其特殊性使得图像置乱方法不能直接应用于音频置乱^[4-6]. 目前的数字音频置乱方法较少,一般采用一维线性映射对音频直接进行置乱变换,这种方法的加密参数较少,并采用了特殊的变换

阵,所以具有固定的变换周期,安全性较差^[7].

基于前期图像置乱的工作基础^[4-6](均为我们团队在图像置乱上的前期研究成果),本文提出一种 n 维空间随机矩阵变换的音频置乱算法.引入 m 进制转换的思想,将一维音序列中每个采样点的音频值提升为 n 维空间坐标值,然后在 n 维空间采用随机变换阵进行变换,最后进行降维处理,还原出一维坐标值,从而完成音频置乱.算法的恢复可通过随机变换阵在 Z_m 上的逆阵一次完成.

1 m 进制的音频变维操作

将音频序列中每个采样点的音频值进行变维操作,变维操作分为升维操作和降维操作.

1.1 升维操作

设音频序列长度为 L ,采样点坐标 $i \in [0, L-1]$,坐标 i 处的音频值 $s_i \in [0, T]$ 为十进制整数,音频最大值 T 限定了 m 进制数的最大长度为 n ,将 s_i 看作一维空间坐标,升维步骤如下.

(1) 设十进制数 $N = s_i$.

(2) 根据 n 为满足 $T/m^n = 0$ 的最小正整数来确定 m 进制数的最大长度 n .

(3) 按照进制转换方法将 N 转换为 m 进制数 $(D_n D_{n-1} \cdots D_1)_m$.

(4) 将 n 个位置的权重系数 $D_n, D_{n-1}, \cdots, D_1$ 作为 n 维空间坐标 $(D_n, D_{n-1}, \cdots, D_1)$.

按步骤(1)~(4),可将一维坐标 s_i 升维为 n 维空间坐标 $(s_{in}, s_{i(n-1)}, \cdots, s_{i1})$.

1.2 降维操作

设 n 维空间坐标为 $(s_{in}, s_{i(n-1)}, \cdots, s_{i1})$,要获得相应的一维空间坐标 s_i ,需做降维处理,步骤如下.

(1) 设 m 进制数的 n 个位置的权重系数 $(D_n, D_{n-1}, \cdots, D_1) = (s_{in}, s_{i(n-1)}, \cdots, s_{i1})$.

(2) 根据进制转换方法,将 m 进制数转换为十进制数 N .

(3) $s_i = N$.

按步骤(1)~(3),可将 n 维空间坐标 $(s_{in}, s_{i(n-1)}, \cdots, s_{i1})$ 降为一维空间坐标 s_i .

2 多维空间音频置乱算法

在现有的采用矩阵变换的置乱算法中,变换阵 $\mathbf{A}$ 多为高维的 Arnold^[3]、Fibonacci-Q 变换阵^[3]和 T 矩阵^[8]等,这些特殊矩阵本身的安全性不高,而且变换后的恢复需要进行正向迭代,恢复代价很高.本文采用文献^[6]的方法生成了随机高维矩阵及其逆

矩阵,以增加矩阵变换的安全性,降低恢复代价.本文的多维空间音频置乱和恢复子算法互为逆过程.

2.1 置乱子算法

置乱子算法的步骤如下.

步骤 1: 音频 Y 中采样点坐标 $i \in [0, L-1]$, L 为采样长度,音频值 $s_i \in [0, T]$, T 为音频最大值.

步骤 2: 根据 n 为满足 $T/m^n = 0$ 的最小正整数来获得 n 的值.

步骤 3: 根据文献^[6]方法,生成一个随机的 n 维变换阵 $\mathbf{A}_{n \times n} \in Z_m$.

步骤 4: for $i=0$ to $L-1$,并按照如下顺序依次执行.

(1) 设十进制数 $N = s_i$,按照进制转换方法得到 m 进制数 $(D_n D_{n-1} \cdots D_1)_m$.

(2) 将 n 个位置的权重系数 $D_n, D_{n-1}, \cdots, D_1$ 作为 n 维空间坐标 $(D_n, D_{n-1}, \cdots, D_1)$,完成升维操作.

(3) 通过高维矩阵

$$\begin{bmatrix} D'_n \\ D'_{n-1} \\ \vdots \\ D'_1 \end{bmatrix} = \mathbf{A}_{n \times n} \begin{bmatrix} D_n \\ D_{n-1} \\ \vdots \\ D_1 \end{bmatrix} \bmod m \quad (1)$$

将 n 维空间坐标 $(D_n, D_{n-1}, \cdots, D_1)$ 转换为 $(D'_n, D'_{n-1}, \cdots, D'_1)$,其中 $\mathbf{A}_{n \times n}$ 是步骤 3 生成的随机 n 维变换阵,且 $|\mathbf{A}_{n \times n}|$ 与 m 互质.

(4) 根据 1.2 节降维操作,将 n 维空间坐标 $(D'_n, D'_{n-1}, \cdots, D'_1)$ 降为十进制数 N' .

(5) $s_i = N'$,即坐标 i 处的音频值被重新赋值.

步骤 5: 保存 Y' ,置乱完成.

2.2 恢复子算法

恢复子算法同 2.1 节置乱子算法类似,限于篇幅,下面仅给出不同点.

(1) 执行 2.1 节的步骤 3,即根据文献^[6]方法生成随机的 n 维变换阵 $\mathbf{A}_{n \times n}$ 的逆阵 $\mathbf{A}'_{n \times n}$, $\mathbf{A}$ 和 $\mathbf{A}'$ 满足 $(\mathbf{A}\mathbf{A}') \bmod m = \mathbf{I}$,其中 $\mathbf{I}$ 为单位阵.

(2) 执行 2.1 节步骤 4 中的第(3)步,即在使用高维矩阵转换(式(1))时,用 n 维变换阵 $\mathbf{A}_{n \times n}$ 的逆阵 $\mathbf{A}'_{n \times n}$ 替换 $\mathbf{A}_{n \times n}$.

3 实验

对 n 维空间随机矩阵变换的音频置乱算法进行了实验.音频样本 test1.wav 为完整的测试音频,见图 1a,其中采样时间为 4.150 ms,采样点数为 183 040,量化精度为 16 b,采样频率为 44 100 Hz.

为了使实验结果更为清晰,截取 test1. wav 音频前 1 597个采样点构成音频样本 test2. wav,见图 1b.

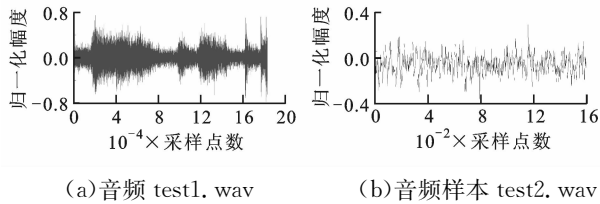


图 1 音频样本波形图

根据文献[6]方法,随机生成了 12 组高维变换阵及其逆矩阵,如图 2 所示.

3.1 置乱及恢复实验

将 test2. wav 各采样点音频值分别转换到 2~4

$$\begin{bmatrix} 29 & 45 \\ 246 & 109 \end{bmatrix} \begin{bmatrix} 23 & 169 \\ 190 & 39 \end{bmatrix} \begin{bmatrix} 49 & 128 \\ 118 & 243 \end{bmatrix} \begin{bmatrix} 209 & 128 \\ 46 & 59 \end{bmatrix} \begin{bmatrix} 5 & 226 \\ 221 & 197 \end{bmatrix} \begin{bmatrix} 17 & 230 \\ 182 & 81 \end{bmatrix} \begin{bmatrix} 255 & 142 \\ 25 & 161 \end{bmatrix} \begin{bmatrix} 33 & 114 \\ 103 & 127 \end{bmatrix}$$

(a) $\mathbf{A}_1$ (b) $\mathbf{A}'_1$ (c) $\mathbf{A}_2$ (d) $\mathbf{A}'_2$ (e) $\mathbf{A}_3$ (f) $\mathbf{A}'_3$ (g) $\mathbf{A}_4$ (h) $\mathbf{A}'_4$

$$\begin{bmatrix} 119 & 148 & 106 \\ 11 & 42 & 253 \\ 254 & 25 & 161 \end{bmatrix} \begin{bmatrix} 17 & 206 & 96 \\ 127 & 207 & 119 \\ 219 & 69 & 226 \end{bmatrix} \begin{bmatrix} 31 & 161 & 221 \\ 142 & 217 & 154 \\ 218 & 243 & 27 \end{bmatrix} \begin{bmatrix} 225 & 156 & 113 \\ 130 & 199 & 48 \\ 0 & 249 & 21 \end{bmatrix} \begin{bmatrix} 213 & 214 & 152 \\ 86 & 35 & 152 \\ 140 & 93 & 127 \end{bmatrix} \begin{bmatrix} 233 & 102 & 232 \\ 238 & 135 & 120 \\ 226 & 83 & 247 \end{bmatrix} \begin{bmatrix} 11 & 234 & 190 \\ 156 & 17 & 46 \\ 12 & 164 & 229 \end{bmatrix} \begin{bmatrix} 147 & 250 & 114 \\ 36 & 129 & 18 \\ 28 & 212 & 141 \end{bmatrix}$$

(i) $\mathbf{A}_5$ (j) $\mathbf{A}'_5$ (k) $\mathbf{A}_6$ (l) $\mathbf{A}'_6$ (m) $\mathbf{A}_7$ (n) $\mathbf{A}'_7$ (o) $\mathbf{A}_8$ (p) $\mathbf{A}'_8$

$$\begin{bmatrix} 225 & 156 & 176 & 221 \\ 161 & 106 & 47 & 14 \\ 35 & 217 & 229 & 23 \\ 138 & 88 & 118 & 91 \end{bmatrix} \begin{bmatrix} 17 & 206 & 96 & 125 \\ 120 & 77 & 15 & 219 \\ 17 & 3 & 222 & 85 \\ 64 & 110 & 140 & 203 \end{bmatrix} \begin{bmatrix} 236 & 79 & 78 & 234 \\ 57 & 123 & 110 & 188 \\ 16 & 220 & 9 & 148 \\ 184 & 203 & 64 & 215 \end{bmatrix} \begin{bmatrix} 177 & 109 & 220 & 150 \\ 137 & 196 & 186 & 98 \\ 56 & 112 & 137 & 4 \\ 99 & 36 & 174 & 253 \end{bmatrix} \begin{bmatrix} 198 & 37 & 156 & 69 \\ 175 & 247 & 83 & 61 \\ 45 & 70 & 128 & 47 \\ 47 & 166 & 121 & 217 \end{bmatrix} \begin{bmatrix} 11 & 191 & 169 & 23 \\ 220 & 93 & 62 & 209 \\ 36 & 216 & 5 & 145 \\ 103 & 49 & 168 & 225 \end{bmatrix} \begin{bmatrix} 245 & 23 & 81 & 0 \\ 56 & 58 & 201 & 246 \\ 196 & 219 & 115 & 238 \\ 244 & 253 & 101 & 35 \end{bmatrix} \begin{bmatrix} 113 & 162 & 241 & 2 \\ 212 & 49 & 157 & 124 \\ 208 & 223 & 64 & 82 \\ 232 & 16 & 65 & 177 \end{bmatrix}$$

(q) $\mathbf{A}_9$ (r) $\mathbf{A}'_9$ (s) $\mathbf{A}_{10}$ (t) $\mathbf{A}'_{10}$ (u) $\mathbf{A}_{11}$ (v) $\mathbf{A}'_{11}$ (w) $\mathbf{A}_{12}$ (x) $\mathbf{A}'_{12}$

图 2 随机生成的高维变换阵及其逆阵($\mathbf{A}_1 \sim \mathbf{A}_4 (m=256)$, $\mathbf{A}_5 \sim \mathbf{A}_8 (m=41)$, $\mathbf{A}_9 \sim \mathbf{A}_{12} (m=16)$)

表 1 在不同 $m, \mathbf{A}$ 下的置乱效果

置乱参数					置乱效果参数		
m	n	变换阵	误码率/%	信噪比/dB	余弦夹角相似度	置乱图图号	运行时间/ms
256	2	$\mathbf{A}_1$	100	-18.221 4	0.574 7	3a	7.620
256	2	$\mathbf{A}_2$	100	-17.923 5	0.535 3	3b	7.290
256	2	$\mathbf{A}_3$	100	-17.866 9	0.562 0	3c	7.137
256	2	$\mathbf{A}_4$	100	-17.928 0	0.565 0	3d	7.061
41	3	$\mathbf{A}_5$	100	-17.864 4	0.550 4	3e	27.421
41	3	$\mathbf{A}_6$	100	-17.930 3	0.572 0	3f	25.779
41	3	$\mathbf{A}_7$	100	-17.876 4	0.578 0	3g	26.223
41	3	$\mathbf{A}_8$	100	-17.948 5	0.562 1	3h	31.044
16	4	$\mathbf{A}_9$	100	-17.927 8	0.673 5	3i	34.458
16	4	$\mathbf{A}_{10}$	100	-18.036 5	0.572 4	3j	35.555
16	4	$\mathbf{A}_{11}$	100	-17.995 0	0.566 5	3k	33.600
16	4	$\mathbf{A}_{12}$	100	-18.125 5	0.545 9	3l	33.687

维空间,对应的 m 为 256、41 和 16,再通过图 2 的正变换阵进行置乱,结果如表 1、图 3 所示. 从表 1、图 3 可看出,原始音频与置乱音频之间的误码率均达到了 100%,信噪比小于 -17 dB,余弦夹角相似度小于 0.58,原始音频与置乱音频的相似度较低,算法运行时间均在毫秒级,置乱代价较低. 置乱结果表明,所提算法具有较好的一次置乱效果.

针对表 1 中置乱变换阵,采用其对应逆阵对置乱音频(限于篇幅,仅对图 3a、3b、3e、3f、3i、3j)进行恢复,结果如表 2、图 4 所示. 可以看出,所提算法能一次性完全恢复置乱音频.

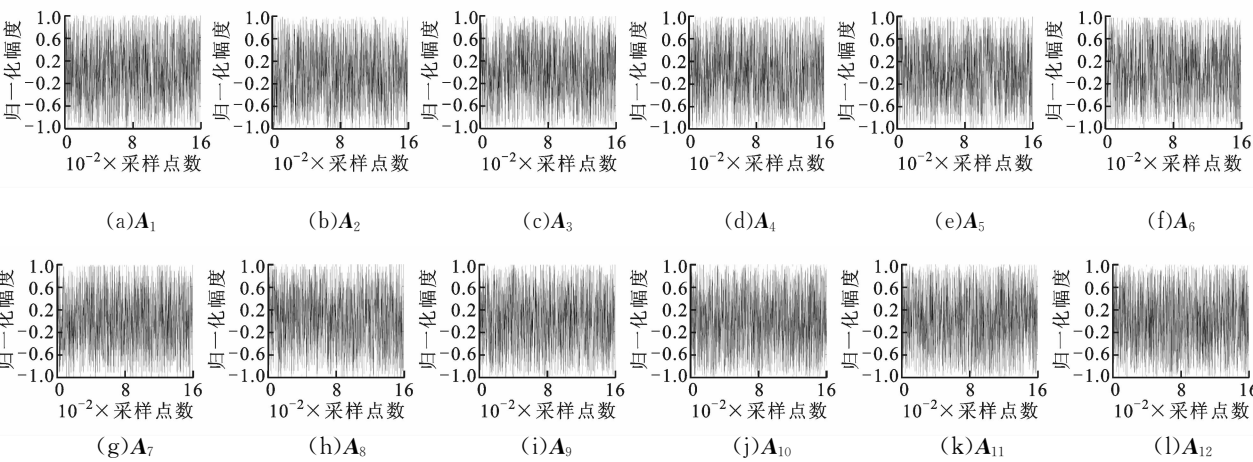


图 3 不同 m, A 对应 n 维空间下的置乱图($A_1 \sim A_4 (m=256, n=2)$,
 $A_5 \sim A_8 (m=41, n=3)$, $A_9 \sim A_{12} (m=16, n=4)$)

表 2 置乱音频未受任何攻击下的恢复效果					
置乱图 图号	逆矩阵	恢复效果参数			恢复图 图号
		误码率/ %	信噪比 /dB	余弦夹角 相似度	
3a	A_1'	0	∞	1	4
3b	A_2'	0	∞	1	4
3e	A_5'	0	∞	1	4
3f	A_6'	0	∞	1	4
3i	A_9'	0	∞	1	4
3j	A_{10}'	0	∞	1	4

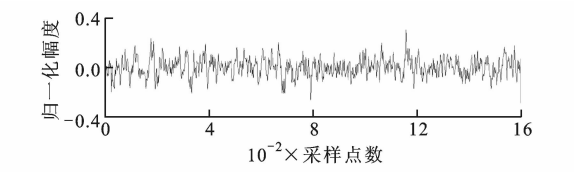


图 4 图 3a、3b、3e、3f、3i、3j 恢复效果图

3.2 攻击后的恢复实验

采用 test1. wav 在 $m=256, A=A_1^{-1}$ 下的置乱音频(图 5a)进行各种攻击实验,并通过逆矩阵 A_1' (图 2b)进行恢复. 置乱音频在各类攻击下的恢复效果如表 3 所示,其中恢复音频和原始音频之间信噪比大于 78 dB,余弦夹角相似度大于 0.78,说明原始音频与恢复音频的相似度较高. 置乱音频(图 5a)经不同攻击后的恢复效果如图 5b~5h 所示,从图中可以看出,置乱音频在受到各种攻击后可基本恢复,说明算法具有一定的鲁棒性.

3.3 同维、不同维空间的置乱效果比较

为了验证同维或不同维空间算法的一次性置乱效果,本文给出了置乱效果的比较,结果如表 4 所

示. 从表中可看出,采用任意 2 个变换阵的置乱音频,其之间的余弦夹角相似度小于 0.58,误码率大于 98%,与表 1 中原始音频与置乱音频的余弦夹角相似度、误码率接近,说明在同维或不同维空间中,对于随机变换阵 A ,算法均具有相近的置乱效果.

表 3 置乱音频在各类攻击下的恢复效果

攻击类型	恢复效果参数		恢复图 图号
	信噪比 /dB	余弦夹角 相似度	
1/2 音频幅值	97.052 3	0.999 0	5b
80 dB 高斯白噪声	90.409 8	0.980 7	5c
任意剪切 1/3 音频长度	87.094 2	0.889 2	5d
音频幅值放大 2 倍	86.464 6	0.887 0	5e
音频幅值缩小 3/5	83.125 1	0.801 5	5f
22 050 Hz 重采样	80.820 0	0.801 0	5g
8 b 重量化	82.130 0	0.821 0	5h

表 4 同维、不同维空间的置乱效果比较

置乱图 1 参数		置乱图 2 参数		置乱图 1 与置乱图 2 的余弦夹角相似度	误码率 /%
n	图号	n	图号		
2	3a	2	3b	0.575 3	100.000 0
3	3c	3	3d	0.578 2	98.058 9
4	3e	4	3f	0.568 4	100.000 0
2	3a	3	3d	0.567 8	100.000 0
2	3b	4	3e	0.568 9	100.000 0
3	3c	4	3f	0.560 4	100.000 0

3.4 与 Fibonacci 变换置乱算法的比较

根据 3.3 节实验,本文算法在同维或不同维空间中对随机变换阵 A 均有相同的置乱效果,因此可任选一个置乱效果与文献[7]方法进行比较,结果如

表 5 所示. 从表中可看出,本文算法无需限定音频长度(即采样点数),比文献[7]方法的置乱效果更好,加密参数更多,并可一次性完全恢复.

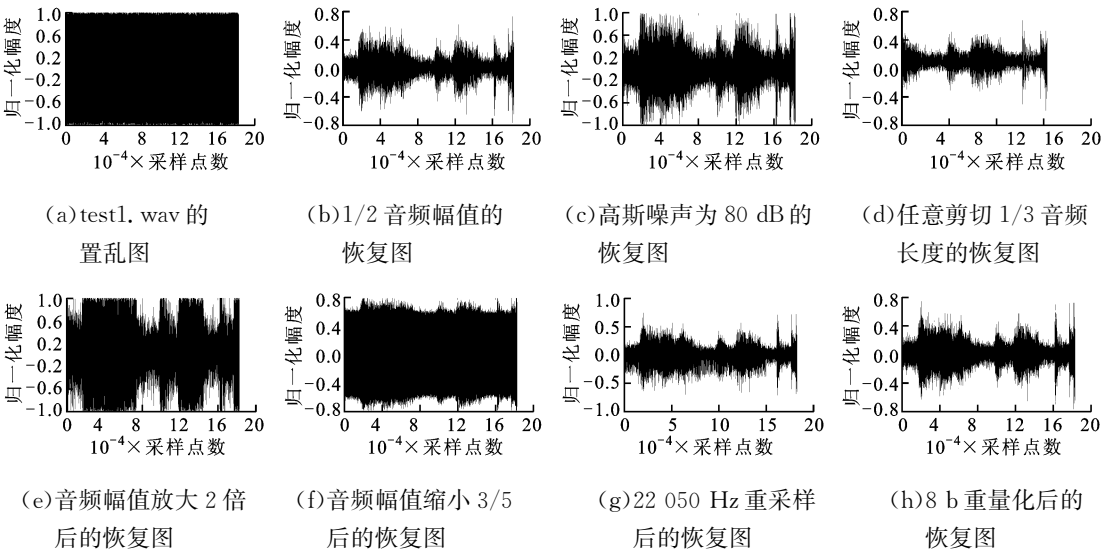


图 5 置乱音频(图 5a)经不同攻击后的恢复效果图

表 5 本文算法与文献[7]方法的置乱效果比较

置乱法	音频长度	误码率 /%	信噪比 /dB	峰值信噪比 /dB	加密参数 数数量/个	恢复次数	恢复情况
本文, $m=16,n=4,A=A_{10}$	不限	100.000 0	-18.036 5	66.414 6	16	1	完全
文献[7], $F_n=987,F_{n+1}=159 7$	1 597	99.937 4	-3.016 9	74.016 8	2	4	完全

注: F_n 、 F_{n+1} 是 Fibonacci 序列中 2 个相邻的数.

4 结 论

本文提出了一种 n 维空间随机矩阵变换的音频置乱算法,该算法包括音频置乱和音频恢复 2 个子算法. 算法的本质是对一维音频进行变维操作,与统一维音频置乱算法相比,其加密参数的组合性更强. 采用随机变换阵进行矩阵变换可以克服特殊变换阵固定周期带来的安全隐患,通过随机变换阵在 Z_m 上的逆矩阵进行一次性恢复,使得解密效率更高. 置乱实验表明,所提算法具有较好的置乱效果,恢复代价比较低,并能抵抗如添加噪声、滤波、剪切、缩放音频等攻击.

参考文献:

[1] 丁伟, 闫伟齐, 齐东旭. 基于置乱与融合的数字图像隐藏技术及其应用 [J]. 中国图形图象学报, 2000, 5 (8): 644-649.

[2] 周海涛, 周詮. 一种压缩域低灰度平滑块数字水印方案 [J]. 西安交通大学学报, 2008, 42 (4): 404-408.

[3] ZHOU Haitao, ZHOU Quan. Digital watermarking scheme with low luminance smooth blocks in discrete cosine transform domains [J]. Journal of Xi'an Jiaotong University, 2008, 42(4): 404-408.

[4] QI Dongxu, ZOU Jiancheng, HAN Xiaoyou. A new class of scrambling transformation and its application in the image information covering [J]. Science in China, Series E: Technological Sciences, 2000, 43(3): 304-312.

[5] 邵利平, 覃征. 基于矩阵变换的图像置乱逆问题求解 [J]. 电子学报, 2008, 36(7): 1355-1363.

[6] SHAO Liping, QIN Zheng. Solution for the inverse

DING Wei, YAN Weiqi, QI Dongxu. Digital image information hiding technology and its application based on scrambling and amalgamation [J]. Journal of Image and Graphics, 2000, 5(8): 644-649.

- problem of matrix transform based image scrambling [J]. Acta Electronica Sinica, 2008, 36 (7): 1355-1363.
- [5] 邵利平, 覃征. 一种基于与(或)逻辑实现的信息分存算法 [J]. 西安交通大学学报, 2006, 40(10): 1078-1082.
- SHAO Liping, QIN Zheng, LI Huan. Information sharing algorithm based on conjunction and disjunction logic functions [J]. Xi'an Jiaotong University, 2006, 40(10): 1078-1082.
- [6] SHAO Liping, QIN Zheng, LI Huan. Scrambling matrix generation algorithm for high dimensional image scrambling transformation [C]// The 3rd IEEE International Conference on Industrial Electronics and Applications. Los Alamitos, CA, USA: IEEE Computer Society, 2008: 1707-1712.
- [7] 李南, 商艳红, 邹建成. 基于 Fibonacci 变换的音频置乱算法 [J]. 北方工业大学学报, 2004, 16(3): 8-11.
- LI Nan, SHANG Yanhong, ZOU Jiancheng. An audio scrambling method based on Fibonacci transformation [J]. Journal of North China University of Technology, 2004, 16 (3): 8-11.
- [8] ZHANG M R, SHAO G C. T-matrix and Its applications in image Processing [J]. IEEE Electronics Letters, 2004, 40(25): 1583-1584.

(编辑 苗凌)